

# Generating Robust and Efficient Networks Under Targeted Attacks

Vitor H. P. Louzada<sup>1</sup>, Fabio Daolio<sup>2</sup>, Hans J. Herrmann<sup>1,3</sup>, and  
Marco Tomassini<sup>2</sup>

<sup>1</sup> Computational Physics for Engineering Materials, IfB, ETH Zurich, Zurich,  
Switzerland.

<sup>2</sup> Faculty of Business and Economics, University of Lausanne, Lausanne, Switzerland.

<sup>3</sup> Physics Department, Federal University of Ceara, Fortaleza, Brazil.

Address for contact: `marco.tomassini@unil.ch`

**Abstract.** The design of efficient large scale networks is a common problem in Operations Research. Some of those, such as electric power grids, transportation systems, communication networks, and others, must maintain their efficiency even after several failures, or malicious attacks. In this work, we outline a procedure that modifies any given network to enhance its robustness, defined as the size of its largest connected component, whilst keeping a high efficiency, which is described in terms of the shortest paths among nodes. We also show that this generated set of networks is very similar to networks optimized for robustness in several aspects such as high assortativity and an onion-like structure.

**Keywords:** complex network, network robustness, network efficiency, onion-like network

## 1 Introduction

In recent years, network studies have attracted a lot of research. As a result, it has been shown that several artificial (e.g. the Internet, electric-grids, etc.) and natural systems (e.g. chemical reaction networks, food networks, gene regulatory networks, etc.) present characteristics that allows one to classify them as Complex Networks. Their structure and the dynamics of phenomena taking place on them have been intensively studied, thanks to the availability of large data sets [6]. An important aspect of a network, especially in fields where network functionality is essential such as transportation, production, power, and biological networks, among others, is the capability to withstand failures and fluctuations in the functionality of its nodes and links. The design of networked infrastructures with these capabilities can be thought of as an optimal design task and their resilience in the face of targeted failures, such as terrorist attacks is still an open problem. An early important work in this field is Albert et al. [1] where the authors showed by numerical simulations that scale-free networks of the Barabási-Albert type [3], while they are robust against random removal of nodes, are much more vulnerable to the removal of nodes according to node's

degree. In other words, in a scale-free graph if the nodes are removed in decreasing order of degree, starting with the most connected ones, then the network falls apart very quickly because those highly connected nodes are the ones that hold the network together.

In the work of Schneider et al. [8], a procedure is described that successfully modifies scale-free networks so that the largest connected component still has a considerable size after several attacks targeted at the most connected nodes. This feature guarantees that there is at least one path connecting a large number of pairs of nodes after the removal and is considered an appropriate definition of robustness. A natural question that follows is the maintenance of network efficiency after attacks, i.e. a network is efficient in this sense if “good paths” among nodes do not cease to exist after several targeted failures. Using a consolidated definition of efficiency for unweighted networks, we propose an optimization procedure that modifies existing networks in order to improve their efficiency under targeted attacks.

This paper is organized as follows. In Section *Methodology*, we present our measures of robustness, efficiency, and a method to optimize a specific characteristic of a network. Then, we show in Section *Results* several comparisons of optimized and unoptimized networks. We highlight the major points of our contribution in Section *Discussion*.

## 2 Methodology

The proposed methodology is an extension of the work of Schneider et. al [8], who used a hill-climbing procedure to optimize a network robustness against targeted attacks. In this work, we modify the approach by adding a simulated annealing strategy [4] to avoid the search getting trapped in local maxima. This procedure is applied to the following objectives: Robustness, Efficiency, and a combined measure of both. We describe first the cost functions that we chose, followed by the optimization procedure.

### 2.1 Robustness

The definition of network robustness might vary according to a specific application. In this work, we call an *attack* the removal of a node of the network, and the robustness we measure by the size of the largest connected component of the network after this removal, as proposed by Schneider et al. [8]. To quantify it, we proceed with a series of targeted attacks (sequential removal of nodes starting from the highest degree after each interaction) and subsequently measure the robustness after each node removal. Hence, we defined the robustness  $R$  of a network as:

$$R = \frac{1}{N} \sum_{Q=1}^N s(Q) , \quad (1)$$

where  $N$  is the network size measured as the number of nodes and  $s(Q)$  is the fraction of nodes in the largest connected component after  $Q = qN$  removals.

## 2.2 Efficiency

One can think of network efficiency as a low cost of communication among its members. In this light, we relate efficiency with the shortest paths between all pairs of nodes, thus following Latora and Marchiori [5] who defined the network efficiency  $E$  as:

$$E = \sum_{\substack{i,j=1 \\ i \neq j}}^N \frac{1}{l_{ij}}, \quad (2)$$

where  $l_{ij}$  stands for the shortest path length between nodes  $i$  and  $j$ . If  $i$  and  $j$  belong to separate connected components of the network, we set  $l_{ij} \rightarrow \infty$  to guarantee a consistent behavior of the cost function.

## 2.3 Integral Efficiency

Keeping in mind that we would like to keep the efficiency of networks after attacks, it is straightforward to modify the definition of  $R$  to account for this. Hence, we define the Integral Efficiency  $IntE$  as

$$IntE = \frac{1}{N} \sum_{Q=1}^N E(Q), \quad (3)$$

where  $E(Q)$  stands for the efficiency of the network after the removal of  $Q = qN$  nodes. In this case, the value of  $E(0)$  is the cost function  $E$  defined in Section 2.2. Choosing this quantity instead of  $E$ , which does not consider nodes removal in its definition, we try to avoid that the shortest paths among nodes will not increase after targeted attacks.

## 2.4 Optimization procedure

In their work, Schneider et al. [8] propose a simple hill-climbing search to modify a network topology in order to optimize the robustness  $R$  whilst keeping the degree of each node fixed. This restriction is often present in the modification of artificial systems, such as electric grids where constructing a receiver for a new power line in a station might be impractical. Hence, only swaps between lines (edges in the graph) are possible. A theoretical consequence of this restriction is that the underlying degree distribution of the network remains unchanged after these swaps. Clearly, if we had no constraints on the degree distribution, we could design the topology starting from scratch with the robustness and efficiency as objectives in mind, obtaining different optimal topologies. However, in this paper we shall follow Ref. [8].

Next, we present an improved version of the optimization approach using simulated annealing [4] and we describe it for any measure  $M$  that changes after link modification:

#### IV

1. **Initial State.** Let  $G(N, E)$  be a network with  $|N|$  vertices and  $|E|$  edges.
2. **Edge swap.** Choose two pairs of edges  $(i, j)$  and  $(k, l) \in E$  randomly and create the network  $G^*$  by deleting the edges  $(i, j)$  and  $(k, l)$ , and adding the edges  $(i, l)$  and  $(k, j)$ .
3. **Acceptance probability.** Calculate the transition probability  $p$  of the system as:

$$p = \begin{cases} \exp\left(-\frac{M(G) - M(G^*)}{T}\right) & \text{if } M(G^*) < M(G) \\ 1 & \text{if } M(G^*) \geq M(G) \end{cases}$$

4. **Comparison.** Draw a random number  $r$  from  $U(0, 1)$ . If  $r < p$  make  $G = G^*$ , otherwise discard  $G^*$ . Return to Step 2.

This approach allows a network  $G^*$  with  $M(G^*) < M(G)$  to be chosen with finite probability. By doing this, global minima could be reached and inferior local minima could be avoided. Furthermore, by decreasing the value of  $T$  according to the amount of edge swaps executed, it is possible to decrease the acceptance ratio of worst networks when an optimum point is close.

In this work, we decrease the temperature as function of the number  $\tau$  of edge swaps, by following the equation:  $T(\tau) = 0.0001 \times 0.8^\tau$ . Variations to this function have shown little effect on the results. The search is stopped when a predefined amount of edge swaps is reached.

### 3 Results

The procedure outlined in Section 2.4 is applied to the cost functions:  $R$  (Robustness as described in Section 2.1),  $E$  (Efficiency as described in Section 2.2), and  $IntE$  (Integral Efficiency as described in Section 2.3), starting from the same set of randomly generated Barabasi-Albert networks. Hence, we created three sets of networks: *Robustness set*, *Efficiency set*, and *Integral Efficiency set*. As a control, we compare to the original set of BA networks which we call the *Unoptimized set*.

The Unoptimized set is composed of 100 networks of  $n = 1000$  nodes and average degree  $\langle k \rangle = 5.95$ . In each of the following plots, the curves represent the optimized sets and the Unoptimized set, for comparison. Each point is the average value of the optimization procedure using 10.000 edge swaps for every network. Data with the final results of the optimization are presented in Table 1.

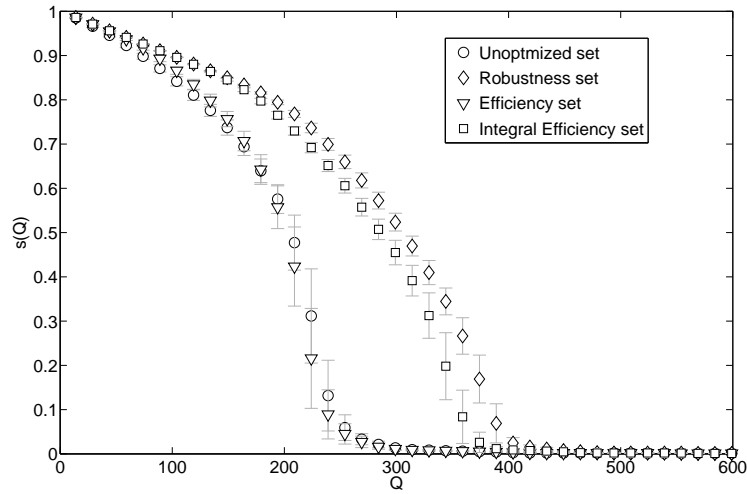
The size of the networks was chosen based on a trade-off between the appearance of topological features such as the scale-free phenomenon, only present in large networks, and computational cost, as the  $IntE$  cost function requires  $O(n^3)$  operations to be calculated. The amount of edge swaps, 10.000, was chosen so that for each optimized set its cost function is already statistically different from the Unoptimized set. It is possible to see that this goal was achieved by comparing the values in bold for columns  $\langle E \rangle$ ,  $\langle R \rangle$ , and  $\langle IntE \rangle$  in Table 1.

To analyze the robustness of each set, a plot of  $s(Q)$  versus  $Q$  is shown in Fig. 1. In this plot, the area below each curve represents the cost function

**Table 1.** Average values, standard deviations in subscripts. Each set comprises 100 networks with  $n = 1000$  nodes.  $\langle k \rangle$  = average degree,  $\langle k^2 \rangle$  = average squared degree,  $\langle cc \rangle$  = average of clustering coefficient,  $\langle r \rangle$  = average assortativity coefficient,  $\langle E \rangle$  = average efficiency,  $\langle R \rangle$  = average robustness,  $\langle IntE \rangle$  = average integral efficiency.

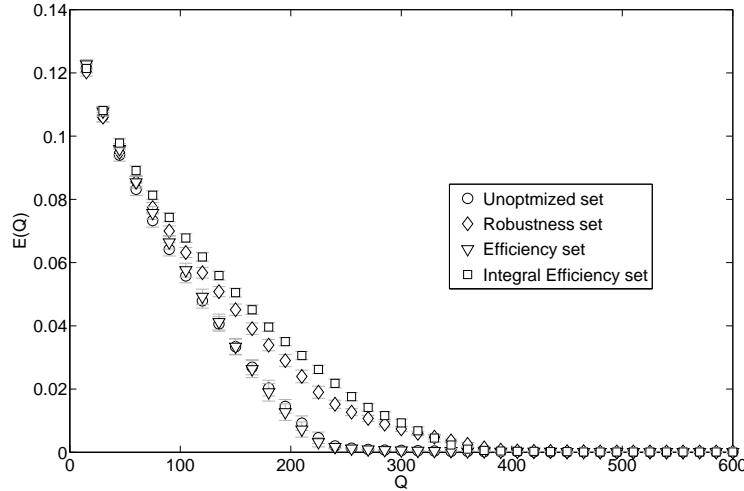
Network Set	$\langle k \rangle$	$\langle cc \rangle$	$\langle r \rangle$	$\langle E \rangle$	$\langle R \rangle$	$\langle IntE \rangle$
Unoptimized	5.95 <sub>0</sub>	0.0242 <sub>0.0033</sub>	-0.085 <sub>0.015</sub>	<b>0.1486</b> <sub>0.0012</sub>	<b>0.1837</b> <sub>0.0053</sub>	<b>0.0308</b> <sub>0.0011</sub>
OptE	5.95 <sub>0</sub>	0.0053 <sub>0.0014</sub>	-0.076 <sub>0.011</sub>	<b>0.1539</b> <sub>0.0015</sub>	0.1826 <sub>0.0056</sub>	0.0310 <sub>0.0012</sub>
OptR	5.95 <sub>0</sub>	0.0200 <sub>0.0027</sub>	0.038 <sub>0.024</sub>	0.1459 <sub>0.0013</sub>	<b>0.2266</b> <sub>0.0055</sub>	0.0372 <sub>0.0012</sub>
OptIntE	5.95 <sub>0</sub>	0.0195 <sub>0.0029</sub>	0.055 <sub>0.026</sub>	0.1456 <sub>0.0013</sub>	0.2268 <sub>0.0052</sub>	<b>0.0391</b> <sub>0.0012</sub>

$R$  for each set. As expected, the Robustness set shows a bigger area (23% of increase), keeping a considerable size on the largest cluster of connected nodes after several attacks. Indeed, Schneider et al. [8] obtained an improvement of almost 75% for this cost function, but by using a much more exhaustive approach: their search stops after 10.000 edge-swaps without increase in  $R$ . Therefore, our results show that it is possible to increase already the network robustness using less computational effort. Moreover, it is interesting to note also that the curve for the Integral Efficiency set has only a slightly smaller area than  $R$ 's one.



**Fig. 1.** Plot of the largest component size after the removal of  $Q$  nodes. The area below each curve is the cost function  $R$ . Each curve represents sets optimized for different cost functions.

In Fig. 2, the cost function  $IntE$  is analyzed through the plot of  $E(Q)$  versus  $Q$ , showing that, as expected, the Integral Efficiency set has the better performance, i.e. the area under the corresponding curve is bigger. Interestingly, the curve referring to the set of networks obtained by optimizing for  $E$  alone shows that they have about the same performance as the unoptimized ones for this cost function (data on Column  $\langle IntE \rangle$  of Table 1).



**Fig. 2.** Plot of the network efficiency  $E(Q)$  after the removal of  $Q$  nodes. The area below each curve is the cost function  $IntE$ . Each curve represents sets optimized for different cost functions.

Another interesting aspect of the work of Schneider et al [8] is the topology obtained by this optimization: a so-called onion-like structure. In this topology, each layer is composed of nodes connected with nodes of the same degree, with few connections between layers. A direct procedure to generate this topology can be found in the work of Wu et al [9].

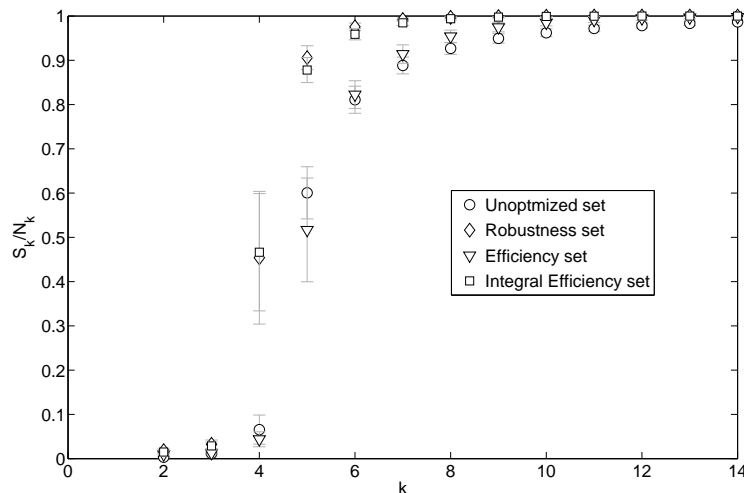
To investigate the presence of an onion-like structure on our optimized sets, three quantities were analyzed. In Fig. 4, we show the  $k$ -core decomposition [2] for several  $k$ , showing that the Robustness and Integral Efficiency sets have several  $k$ -core's or layers, thus confirming a hierarchical structure of the network. The Efficiency set does not present this clear hierarchy, but has more layers than the Unoptimized set.

We also measure the robustness for each layer of a network. To do so, we analyze the subgraph of each network composed of  $N_k$  nodes with degree smaller or equal to  $k$ . In this subgraph,  $S_k$  represents the size of its largest cluster. In Fig. 3, we plot  $S_k/N_k$  for several values of  $k$ . This plot shows that the Robustness and the Integral Efficiency sets present practically the same increase in robustness

with respect to the Unoptimized set. In contrast, the Efficiency set does not show any improvement with respect to the original scale-free unoptimized networks.

Finally, in Fig. 5 we show that the Integral Efficiency set and the Robustness set of networks have the greater assortativity through the plot of Newman's  $r$  coefficient [7]; the Efficiency set is as disassortative as the Unoptimized set.

All this evidence suggests an onion-like structure for the Integral Efficiency set. To provide a visualization of the network structure, some examples of each set were drawn with the  $k$ -core decomposition in Fig. 6.



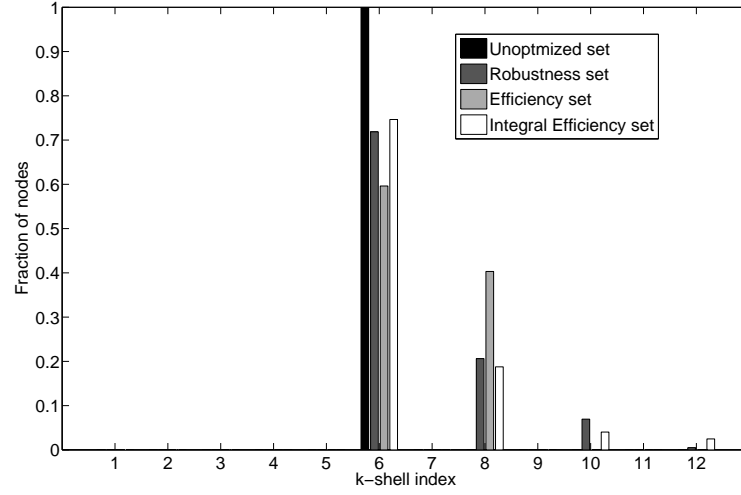
**Fig. 3.** Relative size of the largest component in networks composed of nodes of degree less than  $k$ .

## 4 Discussion

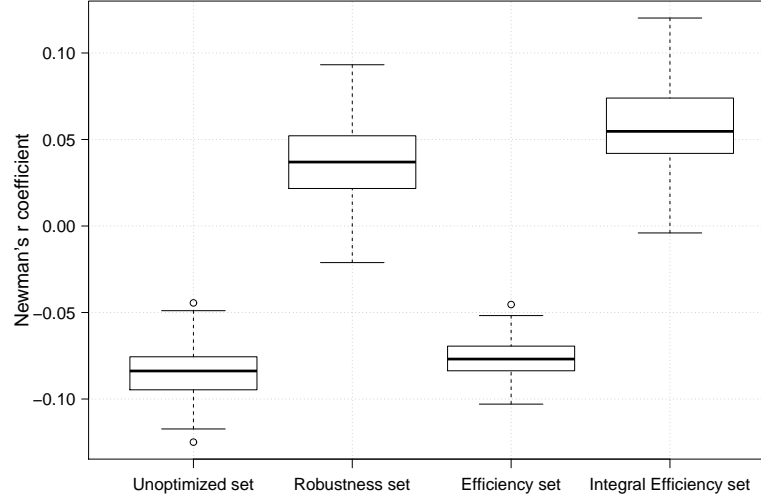
In this paper, by describing a process that optimizes a specific characteristic in any type of network, we are able to create three sets of BA networks with distinguishable features.

Firstly, our results show that the Integral Efficiency set substantially improved efficiency after attacks, compared to the Robustness, Efficiency, and Unoptimized sets. Moreover, this set also sustains a large connected cluster after attacks. Therefore, this cost-function could be used to generate highly robust and efficient networks.

Another important result of our work is that networks optimized for *IntE* also present an onion-like structure. This result suggests that this structure is generically the optimal scale-free net independently of the chosen cost function.

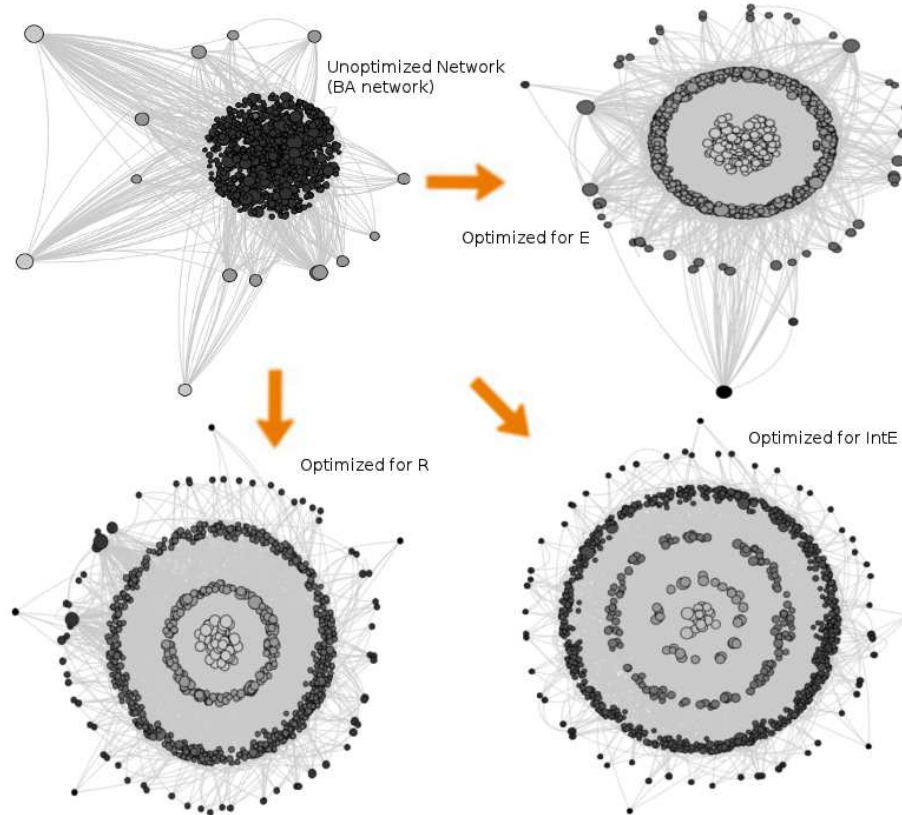


**Fig. 4.** K-core decomposition for several values of  $k$ . It can be seen that the same network optimized for *IntE* presents more layers than the network resulted after the optimization for *R*.



**Fig. 5.** Box-and-whiskers plot of the degree assortativity through Newman's  $r$  coefficient. Thick lines depict the median value; lower and higher hinges gives the 0.25 and 0.75 quantiles, respectively; the whiskers extend to 1.5 times this inter-quantile range. Values outside this range are considered outliers and appear as circle dots in the plot.





**Fig. 6.** Examples of networks belonging to each set. The figures were drawn using the k-core decomposition.

It also helps the design of networks from scratch, as it is possible to construct scale-free networks which present this structure.

It is also interesting to note that the Integral Efficiency set maintains several similarities with the Robustness set, such as: high assortativity, size of the largest cluster after attacks, efficiency after attacks, size of the largest cluster for each degree layer, and a hierarchical structure regarding the k-core decomposition. In fact, the Integral Efficiency set has a slightly better performance on assortativity and efficiency after attacks, while the Robustness set has a better performance on the others.

In the future, we intend to focus on the structures of the three generated sets. The Efficiency set does not present an onion-like structure, remaining unclear if this optimization could lead to a different structure. The Integral Efficiency set might have a hidden feature that differentiates it from the Robustness set. By finding a typical structure of optimized networks, new networks could be designed from scratch with a desired feature. Also, we would like to investigate other cost functions that might lead to onion-like structures, and the case of weighted networks, as they are closer to real applications.

**Acknowledgments.** Authors would like to thank the Swiss National Science Foundation under contract 200021 126853 and CNPq for financial support, and the Cuttlefish software for the network visualization tool.

## References

1. Albert, R., Jeong, H., Barabasi, A.L.: Error and attack tolerance of complex networks. *Nature* 406, 378–382 (2000)
2. Alvarez-Hamelin, J.I., Dall’Asta, L., Barrat, A., Vespignani, A.: k-core decomposition: a tool for the visualization of large scale networks. *Advances in Neural Information Processing Systems* 18(41) (Oct 2005), <http://arxiv.org/abs/cs/0504107v2>
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* 286, 509–512 (1999)
4. Kirkpatrick, S., Gelatt, C.D., Vecchi, P.: Optimization by simulated annealing. *Science* 220, 671–680 (1983)
5. Latora, V., Marchiori, M.: Efficient behavior of small-world networks. *Phys. Rev. Lett.* 87(19), 198701 (2001)
6. Newman, M.E.J.: *Networks: An Introduction*. Oxford University Press, Oxford, UK (2010)
7. Newman, M.: Assortative mixing in networks. *Phys. Rev. Lett.* 89(20), 208701 (2002)
8. Schneider, C.M., Moreira, A.A., Jr., J.S.A., Havlin, S., Herrmann, H.J.: Mitigation of Malicious Attacks on Networks. *Proc. Natl. Acad. Sci. USA* 108(10), 3838–3841 (Mar 2011), <http://arxiv.org/abs/1103.1741v1>
9. Wu, Z.X., Holme, P.: Onion structure and network robustness. *Phys. Rev. E* 84(026106) (Aug 2011), <http://arxiv.org/abs/1108.1841v1>